

### **SPECIFICATION**

*Please amend paragraph [0027] of the specification as follows:*

The IPIC 103 module can interface to the bus on one side and a high speed interface, such as a HiGig™ interface, on the other side. The high speed bus can be, for example, ~~[[is]]~~ a XAUI interface, providing a total bandwidth of 10 Gbps. The CMIC 104 block is the gateway to the host CPU. In ~~[[it's]]~~ its simplest form it provides sequential direct mapped accesses between the CPU and the network device. The bus interface may be a 66 MHz PCI. In addition, an I2C (2-wire serial) bus interface may be supported by the CMIC, to accommodate low-cost embedded designs where space and cost are a premium.

*Please amend paragraph [0034] of the specification as follows:*

Portions of the ingress and processing elements of the network device, according to one embodiment, are illustrated in FIGS. 2 and 3. FIG. 2 illustrates several buffers 201-1 through 201-4 for receiving packet data. The buffers pass the data to the cell assembler 202 and are then passed to the Weighted Random Early Detection (WRED) 203 module to provide congestion avoidance by dropping packets as needed based on IP precedence. The data is then passed to a ~~[[for]]~~ cyclic redundancy check (CRC) 204 module to detect data transmission errors. The data is subsequently passed to a lookup 205 module and then to ingress buffers 206-1 and 206-2. Thereafter, the data passes from the ingress buffer 301-1, in FIG. 3, to a drop filter 302 that may drop the packet based on programmed criteria. The packet data then passes to an arbiter 303, that has its own random access memory 306. The arbiter controls access to the memory channels 305-1 and 305-2 where packet data is stored. The arbiter communicates with a free cell pointer module 304 that provides a pointer to the next available free cells in the memory. The arbiter also is in communication with the egress queue 308 and egress buffer 311 modules. The egress buffer receives the packet data when it is ready to be sent out on the CPE interface. The egress queue module is also in communication with a scheduler 309 that schedules which packets and in what order they are sent out. The scheduler also communicates with a weighted fair queue 310 module to assist in making scheduling decisions, where those decisions result in specific read requests being sent to the arbiter.

*Please amend paragraph [0036] of the specification as follows:*

In one embodiment of the present invention, a Layer 3 table 410 is used as a routing table (step 1). A Longest Prefix Match (LPM) table 420 is used for longest-prefix matching (step 2) to support the ECMP. The entries in the L3 table are grouped to support the multiple paths. Thus for a given IP address, a longest prefix match is made through the LPM table. In the LPM table, at the entry found through the longest prefix match is a ~~[[filed]]~~ field called the count field. The count field is populated based on the number of equal cost paths for a particular IP route. For example, if the count was “4”, that would mean that ~~[[the]]~~ there are four paths are calculated to be of equal cost for that packet to the destination IP address.

*Please amend paragraph [0039] of the specification as follows:*

The process is also illustrated, according to at least one embodiment, in FIG. 5. An L3 destination search is begun, in step 500, and it is determined whether the destination IP address ~~[[in]]~~ is found in the L3 table, in step 501. The LPM table is searched, step 502, and a determination is made whether the destination IP address is found therein, in step 503. If not, the next pointer is determined, step 504, and the process continues iteratively until the address is found, steps 504 and 505 or until all IP addresses are exhausted. In some embodiments, this is only eight iterations. The L3 table index is determined from the LPM table, step 506, and the next hop destination MAC address and the egress port number are obtained, steps 507-509.

*Please amend paragraph [0063] of the specification as follows:*

The HTLS format may be translated into other formats, with the tagging occurring when the packet arrives at the chip and then stripped off on the uplink port. The chip provides the wrapper itself and tables and registers are provided to support HTLS. Double tagging occurs when a packet is sent out with two tags. In HTLS, all packets within the chip have two tags. In addition, a different VC label may be assigned to a packet. The VC label may be assigned by default on a per port basis or the FFP may be ~~[[sued]]~~ used to classify the packet and assign a new VC label for packets coming in from the same port or path. Thus, the VC label information is also carried on top of the double tags inside the chip. On egress, based on the VC label and information in the register, the packet is sent out with one label or two labels in HTLS.